AFRL-SR-AR-TR-10-0190

REPORT DOCUMENTATION PAGE

1. REPORT OATE *(DD-MM-YYYY)*
10-02-2010

2. REPORT TYPE
Final Report

3. DATES COVEREO *(From - To)*
April 15,2009 to Feb 14, 2010

4. TITLE AND SUBTITLE
Compressive Hyperpsectral Imaging and Anomaly Detection

5a. CONTRACT NUMBER
FA9550-09-C-0121

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Stanley Osher, Wotao Yin, Kevin Kelly, Pradeep Thiyanaratnam and Susan Chen

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Level Set Systems
1058 Embury Street
Pacific Palisades, CA 90272

8. PERFORMING ORGANIZATION REPORT NUMBER
1A-2010

9. SPONSORING/MONITORING AGENCY NAME(S) ANO AOORESS(ES)
Air Force Office of Scientific Research
875 Randolph Street, Room 3112
Arlington, VA 22203

10. SPONSOR/MONITOR'S ACRONYM(S)
AFOSR

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
LSL0003Z

12. OISTRIBUTION/AVAILABILITY STATEMENT
Unlimited

20100602216

13. SUPPLEMENTARY NOTES

14. ABSTRACT

We have developed and tested state-of-the-art target detection/template matching methods based on L1 minimization. Given the spectral signature of a material, we are able to identify the pixels in a hyperspectral image, even for very noisy data, that contains the material. The speed of our unmixing algorithm is now much faster than any previous methods. We have also expanded the use of the Bayesian dictionary learning and sparse reconstruction method by utilizing spatial inter-relationships between different components in images and incorporating sparsity of spectral vectors in terms of sparse representation by endmembers into recoonstruction.

15. SUBJECT TERMS

compressive, hyperspectral anomaly, template matching, unmixing, sparse reconstruction, L1, endmember, Bayesian dictionary learning

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Stanley Osher
U	U	U	UU	19	19b. TELEPHONE NUMBER *(Include area code)* 310-573-9339

STTR Final Report
STTR Topic # AF-08-BT24
Proposal #F08B-T24-0062
Contract #FA9550-09-C-0121
Compressive Hyperspectral Imaging and Anomaly Detection

Level Set Systems, Rice CAAM Group, Rice Kelly Lab

April 15, 2009 to February 14, 2010

1 Summary

Level Set Systems and the Rice CAAM group have continued to test the state of the art target detection/template matching method based on L1 minimization. Given the spectral signature of a material, we are able to identify the pixels in a hyperspectral image(HSI) that contains the material. A new data set(from RIT) was used to test the accuracy of the algorithm with very good results. The speed of our unmixing algorithm has been improved and a comparison with the standard nonnegative least squares is given. We have also expanded the use of the Bayesian dictionary learning and sparse reconstruction method by utilizing spatial inter-relationships between different components in images and trying to incorporate sparsity of spectral vectors in terms of sparse representation by endmembers into reconstruction.

2 Improved Unmixing

A hyperspectral image spectral can be composed of a relatively small number of materials. Different materials have different hyperspectral signatures, referred to as endmembers. Due to the usual low resolution of HSI, the pixels often contain a mixture of materials. However, it is often the case that only a few materials are in any given pixel. Unmixing is the process of determining which materials/endmembers are in a given pixel. A hyperspectral image is represented by a three-dimensional data cube M. There is an image for every band in the hyperspectral cube. Each image has m rows and n columns. There are b spectral bands in the data cube. Formally,

$$M \in \mathbb{R}^{m \times n \times b}, \quad M \geq 0.$$

Let $f \in \mathbb{R}^b$ be one of the pixels of the data cube M, and $E = [e_1 \; e_2 \; \cdots \; e_K]$, where e_i are the endmembers(either given or computed). We will assume that a pixel is composed of a linear combination of the endmembers, i.e.,

$$f = \sum_{i=1}^{K} x_i e_i = Ex.$$

If we assume that only a few endmembers are contained in a pixel, then we want the coefficient vector x to be sparse. Additionally, we require $x \geq 0$, since there can only be a positive amount of a material in a pixel.

Following [1], the original ℓ_1-minimization model we used was

$$\min_x |x|_1 + \frac{\lambda}{2}||Ex - f||_2^2, \quad \text{s.t. } x \geq 0.$$

Using the Split Bregman method [2] gives the following algorithm:

```
Initialization $x^0 = b^0 = d^0 = 0, f^0 = f$. We define $\gamma = 1/\lambda$ ;
for $k \leftarrow 1$ to $nOuter$ do
    for $r \leftarrow 1$ to $nInner$ do
        $b^{n+1} = b^n + a^n - d^n$;
        $x^{n+1} = \left(E^T E + \gamma I\right)^{-1} \left(E^T f^k - b^{n+1} + d^n\right)$;
        $d^{n+1} = \max\left(a^{n+1} + b^{n+1} - \mu, 0\right)$;
    end
    $f^{k+1} = f^k + f - Ex^{n+1}$ ;
end
```

Numerically, the above algorithm gives essentially the same solution as the nonnegative least squares problem $min_x \; ||Ax - f||^2 \; s.t. \; x \geq 0$. However, the nonnegative least squares implementation in Matlab was significantly faster than the above algorithm. In order to speed up the computation, we consider a different implementation of the following model [3]:

$$\min_x \; \eta||x||_1 + ||Ex - f||^2, \quad \text{s.t. } x \geq 0,$$

and η is a positive constant. The above is equivalent to the the following problem

$$\begin{array}{l} \min_{x,d} \; \eta \sum x + ||Ex - f||^2; \\ x = P(d), \end{array} \tag{1}$$

where P is a component wise operator defined as

$$P(\alpha) = \begin{cases} \alpha & \alpha > 0 \\ 0 & \text{otherwise,} \end{cases} \tag{2}$$

The constrained problem (1) is replaced by the following sequence of unconstrained problems:

$$\begin{array}{l} d^{k+1}, x^{k+1} = \min_{x,d} \; \lambda\eta \sum x + \lambda||Ex - f||^2 + ||x - P(d) - b^k||^2, \\ b^{k+1} = b^k + P(d^{k+1}) - x^{k+1}, \end{array} \tag{3}$$

where λ is a user defined parameter.

The first subproblem can also be split into two, giving

$$\begin{array}{l} d^{k+1} = \min_d \; \lambda\eta \sum x + \lambda||Ex^k - f||^2 + ||x^k - P(d) - b^k||^2, \\ x^{k+1} = \min_x \; \lambda\eta \sum x + \lambda||Ex - f||^2 + ||x^k - P(d^{k+1}) - b^k||^2, \\ b^{k+1} = b^k + P(d^{k+1}) - x^{k+1}, \end{array} \tag{4}$$

Each subproblem can now be solved exactly using the iterations:

$$\begin{array}{l} d^{k+1} = P(x^k - b^k), \\ x^{k+1} = (\lambda E^T E + I)^{-1}(P(d^{k+1}) + b^k + \lambda E^T f - \lambda\eta I), \\ b^{k+1} = b^k + P(d^{k+1}) - x^{k+1}, \end{array} \tag{5}$$

2.1 Experimental Results

We applied the improved algorithm to a dataset of Moffet Field, CA obtained from NASA [4]. It has spatial dimensions of 500x614 and 224 spectral bands. Six endmembers were chosen with parameters $\lambda = \frac{80}{||A^T A||_2}$ and $\eta = 1$. Results are given in Figures 1-6. The improved algorithm also has a significant edge over the nonnegative least squares method of Matlab in computational time. The table below gives comparisons of the computational time for 4 different hyperspectral data sets in addition to the Moffet Field data. The other data sets include the URBAN set from [5], and 3 data sets provided to us by Major Fay Spellerberg of the USAF.

Data	Spatial Dim.	Spec. Bands	# of endmembers	Time for NNLS(s)	Time for L1 unmixing(s)
Moffet Field	500x614	224	6	246	42
URBAN	307x307	163	6	76.4	12.2
AF 011	573x256	210	4	102.5	19.1
AF 012	573x256	210	4	98.7	19.0
AF 016	573x256	210	4	109.9	19.0

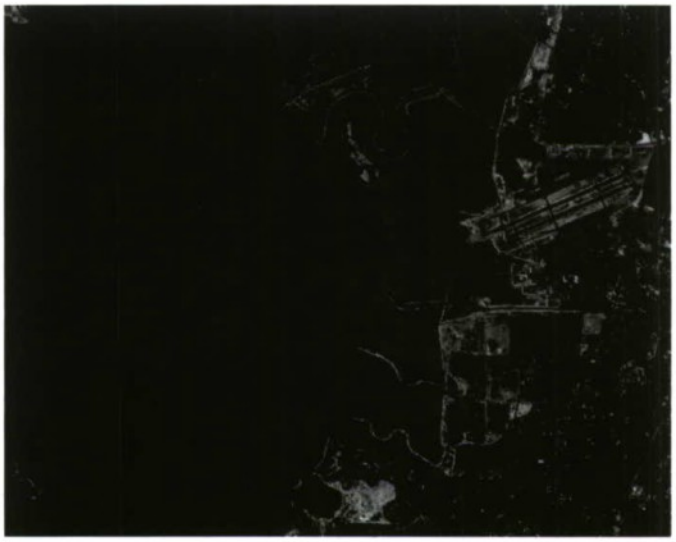


Figure 1: Unmixed data

3 Anomaly/Target Detection

We have further tested out anomaly/target detection algorithm, which is summarized as follows. Given a HSI with NxN pixels and M spectral bands, we wish to locate the positions of pixels that correspond to a given spectral signature f, which also has M spectral bands. We rearrange A as an $M \times N^2$ matrix, where generally $M < N^2$. The signals a_i are the columns of this spectral matrix A and correspond to each pixel in the image.

Our goal is to find $u \in R^{N^2}$ by solving the constrained minimization problem

$$\begin{aligned} &u = argmin\,|u|_1 \;\; s.t. \;\; \|Au - f\| < \delta, \\ &u \geq 0, \end{aligned} \tag{6}$$

where δ is a measure of the noise in the system.

To solve this, we apply Bregman iteration [6, 7], by solving a sequence of unconstrained minimization problems.

$$u^{n+1} = argmin\left(\mu\,|u|_1 + \frac{\lambda}{2}\,\|Au - f^n\|^2\right) \tag{7a}$$

Figure 2: Unmixed data

$$f^n = f^{n-1} + f - Au^{n-1} \tag{7b}$$

for $n = 1, 2, ...$, with $u^0 = 0$. The constant λ is usually chosen around $\lambda = \frac{100}{\|A^T A\|}$. $\|Au - f^n\|^2$ monotonically decreases to zero and u^n converges very quickly to a solution of (6) with $\delta = 0$, see [7, 6].

It now becomes a matter of solving (7a) and (6) efficiently. We propose two recently developed algorithms (1) Split Bregman[2] for (7a) and/or (6), linearized Bregman for (6) [8, 9, 10, 7]. The idea behind both of these is quite simple. There are two simple minimization problems to be solved. A combination of these solvers will converge to the desired solution, as we described before. To solve

$$argmin\left(\mu\,|u|_1 + \frac{1}{2}\|u - f\|^2\right) \tag{8}$$

we have the following well known shrinkage formula

$$u_i = shrink(f_i, \mu) = \begin{cases} f_i - \mu & \text{if } f_i > \mu \\ 0 & \text{if } |f_i| \leq \mu \\ f_i + \mu & \text{if } f_i < -\mu \end{cases} \tag{9}$$

Moreover, if we add the constraint that $u_i \geq 0$, then

$$u_i = shrink^+(f_i, \mu) = \begin{cases} f_i - \mu & \text{if } f_i > \mu \\ 0 & \text{if } f_i \leq \mu \end{cases} \tag{10}$$

To solve

$$argmin\left(\frac{\lambda}{2}\|Au - f\|^2 + \frac{1}{2}\|d - u\|^2\right) \tag{11}$$

for a fixed vector d, we have

$$u = \left(\lambda A^T A + I\right)^{-1}\left(\lambda A^T f + d\right) \tag{12}$$

Figure 3: Unmixed data

The idea behind split Bregman is s follows: We replace the problem (7a) by a sequence of approximations generated by Bregman iteration:

$$\left\{ \begin{array}{l} \left(d^{k+1}, U^{k+1}\right) = argmin \mu \left|d\right|_1 + \frac{\lambda}{2} \left\|Au - f\right\|^2 + \frac{1}{2} \left\|d - U - b^k\right\|^2 \\ b^k = b^{k-1} + U^k - d^{k-1} \end{array} \right. \tag{13}$$

The steps used in the solution for (13a) and (13b) involve splitting

$$\left\{ \begin{array}{l} U^{k+1} = \left(\lambda A^T A + I\right)^{-1} \left(\lambda A^T f^n - b^k + d^k\right) \\ d^{k+1} = shrink\left(U^{k+1} + b^{k+1}, \mu\right) \end{array} \right. \tag{14}$$

U^k approaches u^{n+1} monotonically, $\left\|U^k - u^n\right\| \searrow 0$, and of course, $\left\|d - U^k\right\| \searrow 0$. Thus we use an inner iteration to obtain the sequence U^k,d^k, which converges to the updated u. We then update using (7) to get f^{n+1} and repeat the inner iteration to get u^{n+2}. This procedure is very efficient. The number of of inner iterations needed is problem dependent, but usually between 5 and 10.

Alternatively, we may use the linearized Bregman approach [8, 9, 10, 7] to solve (6) directly. This involves a very simple 2 line code and has the advantage that we need not compute the matrix inverse appearing in (14a). However, it is a bit slower for bigger matrices A. The entire algorithm is as follows:

$$\left\{ \begin{array}{l} u^{k+1} = \delta shrink\left(v^k, \mu\right) \\ v^{k+1} = v^k + A^T\left(f - Au^{k+1}\right) \end{array} \right. \tag{15}$$

for $v^0 = 0$, $k = 0, 1, ...$ with $\frac{\delta}{2} AA^T < I$, $\delta > 0$.

For μ sufficiently large, this converges to a solution of (6), see [11]. However, in general the solution u satisfies

$$u = argmin\left(\left|u\right|_1 + \frac{1}{2\delta\mu}\left\|u\right\|^2\right) \tag{16}$$

such that $Au = f$.

Figure 4: Unmixed data

3.1 Experimental Results

3.1.1 RIT Data

We obtained some HSI data from Rochester Institute of Technology(RIT) [12] to test our target detection algorithm. The first set of targets consisted of 6 pieces of cloth, made of nylon or cotton, and of different sizes and colors. We ran the target detection algorithm for each of the 6 cloth targets, and used the ground truth location as the spectral signature for each target. A 80x80x126 subset of the whole data set, which had size 280x800x126, was used for the computation. Figure 9 shows the ground truth locations of the 6 pieces of cloth and Figure 10 shows the computed locations of the targets. The locations were exactly computed with no false pixels identified.

A second set of targets consisting of 3 vehicles was also used to test the algorithm. Each vehicle was detected in a separate run of the algorithm, and the ground truth locations of the targets were used as the spectral signatures. Similar to the first test, we used a subset of size 80x80x126 for the experiment, and the locations of the vehicles were exactly detected with no falsely identified pixels. Figure 7 shows the ground truth locations of the 3 vehicles and Figure 8 shows the computed locations of the targets.

3.1.2 Detecting Targets on the Whole Data Set

Running our target detection algorithm on the entire 280x800x126 HSI would have been prohibitive due to large memory requirements. We are currently testing methods to overcome this limitation. One such method is to break up the data cube into pieces spatially, and run the algorithm on each piece. However, numerical experiments revealed that if the subset of the data cube does not contain the target, the algorithm may falsely identify some pixels as targets. The reason for this is that the solution to (6) tends to produce solutions $u \neq 0$, even though that is the ideal solution if the target f is not present in the data cube. One possible solution to this problem is to slightly alter the matrix of pixels A, by adding a column f_δ at the end. $f_\delta = f + \delta$, where δ is some noise. Then the new matrix becomes $\hat{A} = [A : f_\delta]$ and has size $M \times (N^2 + 1)$.

Figure 5: Unmixed data

Then we solve the altered problem

$$\begin{aligned} &\hat{u} = argmin \left|\hat{u}\right|_1 \;\; s.t. \;\; \left\|\hat{A}\hat{u} - f\right\| < \delta, \\ &\hat{u} \geq 0, \end{aligned} \tag{17}$$

If A does not contain the target f, the solution should be $\hat{u} = (0, \dots, 0, \alpha)$, where $\alpha > 0$. We then take the solution to the original problem as $u_i = \hat{u}_i$, $i = 1, \dots, N^2$. If A does contain the targets, the addition of f_δ to the end of A does not effect the identification of the real targets. We tested this on one of the cloth targets, result shown in Figure 11, and on one of the vehicles, shown in Figure 12. As seen in the figures, the targets were detected exactly with no falsely identified pixels. Each of the subsets were of size 40x40x126, and a total of 140 calculations were required to process the entire 280x800x126 data cube. The calculated threshold value(described in the section below) for these was $\epsilon = 0$, so falsely identified pixels were not a problem for this particular data set. Noise was also not a problem for this data set, so we set $\delta = 0$.

3.1.3 Thresholding

Our previous experiments have shown that the number of falsely identified pixels can be reduced by thresholding the values of $u_i > 0$. Therefore, we consider pixel i as a falsely identified target if $u_i < \epsilon$ for some threshold value $\epsilon > 0$. So far, we have determined the threshold value ϵ by trial and error, but we present a method for computing it automatically. Let $u^s = \{u_i : u_i > 0\}$. Next, sort the values of u^s in descending order. Our numerical experiments have shown that the values u_i of the falsely detected pixels are generally much smaller that the values of the correctly identified pixels. This suggest that the threshold should be where the largest jump is in u^s. As an example, we consider the real HSI data provided to us by NGA, which was analyzed in our second report. More specifically, we look at the plot of u^s for the gravel target, shown in Figure 13. It is clear where the biggest jump in u^s is and suggests that the threshold value in this case should be 0.0079.

Figure 6: Unmixed data

4 Bayesian Dictionary Learning for Sparse Image Representations and Reconstruction

4.1 Summary

We applied the recent Bayesian dictionary learning method [13] to reconstruct hyperspectral images from very few observations. The Bayesian dictionary learning method models a hyperspectral image as the sparse sum of a set of dictionary atoms, which are assumed to follow certain distributions and learned by statistical inference from partial observations of the hyperspectral image. The size of dictionary can be inferred nonparametrically. For reconstructing hyperspectral images from very few samples, no prior knowledge of the noise variance needs to be assumed, and the noise variance can also be non-stationary. We have utilized spatial inter-relationships between different components in images and tried to incorporate sparsity of spectral vectors in terms of sparse representation by endmembers into reconstruction. Numerical experiments using real hyperspectral were performed, with very good results.

4.2 Sparse Image Representation

Image reconstruction and analysis are based on how images are represented. In the standard representation, a natural signal is treated as an array of pixels in space or time. This is convenient for digitally sampling, display or playback. However, it is inefficient for many reconstruction and analysis tasks.

A more meaningful representation shall describe the useful characteristics of the signal: for reconstruction from noisy measurements, the representation should efficiently separate signal and errors; for compression, the representation should capture a large part of the signal with a sparse or compressible set of coefficients; for analysis such as decomposition and recognition, the representation should highlight salient features. They seem to be different goals but they all look for a sparse representation of features.

One way of such representation involves the choice of a dictionary, which is the set of elementary signals C or atoms C used to decompose the signal. Consider a signal $x \in \mathbb{R}^n$ and a fixed dictionary $D = [d^1\ d^2\ \cdots\ d^M]$,

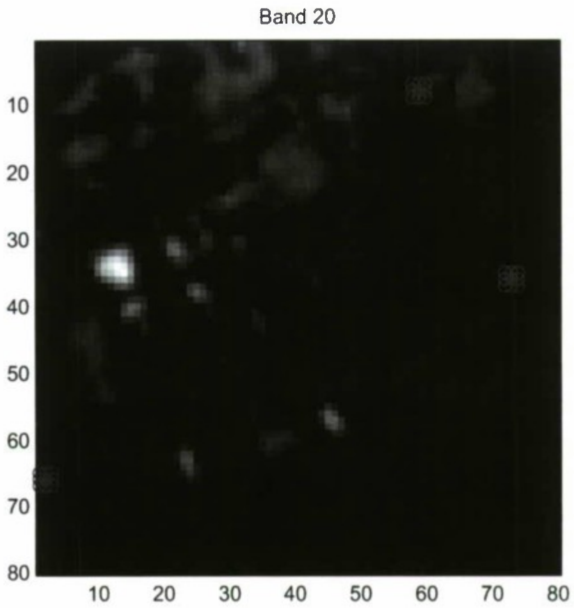


Figure 7: Ground truth locations of the 3 vehicles.

where each $d^m \in \mathbb{R}^n$. We wish to have such a D that x may be well approximated by $\hat{x} = D\alpha$. In the simplest case the dictionary is orthogonal. Examples include the discrete cosine basis and various wavelets based bases. They have been thoroughly studied and widely considered in applications because they are easy to analyze and they have fast numerical implementations. However, they are over-simplistic for certain real data including hyperspectral imagery. To find sparser, meaningful representations for more signals, researchers have recently developed non-orthogonal dictionaries, some of which are overcomplete (i.e., $M > n$) and/or trained, as opposed to analytic.

The earliest major work in dictionary training is due to Olshausen and Field [14], who trained an overcomplete dictionary for sparsely representing small image patches of a set of natural images. Remarkable results were obtained from a simple algorithm, namely, the atoms in the trained image were very similar to the simple cell receptive fields in early vision. What is remarkable in this finding is that sparsity plays a key role in biological visual behavior. It suggests the potential of sparse representation in uncovering fundamental features in complex signals.

Inspired by Olshausen and Field's and others' work [14, 15, 16, 17], especially the recent work by Lawrence Carin's group [18, 13], we wish to impose that the coefficients α_i in the representation $\hat{x} = D\alpha$ are sparse. With a proper D, the computation of sparse α is robust to noise and numerically tractable even when x is partially or indirectly observed via a small number of measurements (as arising in problems such as inpainting, interpolation and compressive sensing). The recent work [13] is different from all previous work in the following aspects. First, when D is given, previous work computes α as a point estimate (i.e., a vector) but [13] returns a posterior distribution through statistical inference. Second, to return the point estimate of α, previous work uses algorithms such as basis pursuit and orthogonal matching pursuit, for which the parameters and stopping criteria are defined based on assumed knowledge such as noise variance or the sparsity of true α. However, the noise variance can be inferred in [13]. Last, to learn the dictionary D, its size M is fixed but the statistical inference method does not require this assumption.

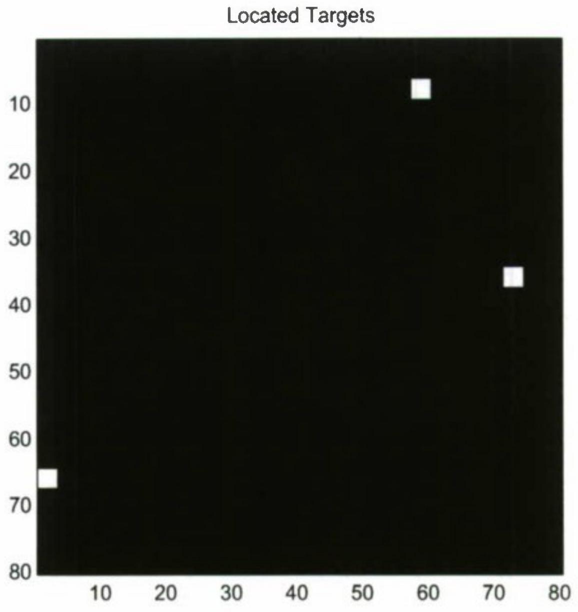


Figure 8: Found locations of the 3 vehicles.

4.3 The Dictionary Model

Specifically, the method we use is based on the Beta process [18] applied to the model $x^i = D\alpha^i + \epsilon^i$ [13], where each $x^i \in \mathbb{R}^n$ is a hyperspectral image, $D = [d^1\ d^2\ \cdots\ d^M] \in \mathbb{R}^{n\times M}$ is a dictionary, $\alpha^i \in \mathbb{R}^M$ is a sparse vector, and $\epsilon^i \in \mathbb{R}^n$ is noise. A hierarchical model, which is a dependence graph of random variables, is assumed:

$$\begin{aligned} d^m &\sim \mathcal{N}(0, P^{-1}I_P), && (18)\\ \alpha^i &\sim z^i \odot s^i, && (19)\\ z^i &\sim \Pi_{m=1}^{M}\text{Bernoulli}(\pi_m) && (20)\\ \pi_m &\sim \text{Beta}(a_0/K, b_0(K-1)/K) && (21)\\ s^i &\sim \mathcal{N}(0, \gamma_s^{-1}I_K) && (22)\\ \epsilon^i &\sim \mathcal{N}(0, \gamma_\epsilon^{-1}I_P) && (23)\\ \gamma_s &\sim \Gamma(c_0, d_0) && (24)\\ \gamma_\epsilon &\sim \Gamma(e_0, f_0). && (25) \end{aligned}$$

Here each atom d^m follows a Gaussian distribution. Each vector α^i is the Hadamard (component-wise) product of a 0/1 Bernoulli vector z^i and Gaussian vector s^i. z^i defines which the atoms of the dictionary are used to represent image x^i, and w^i contains the representation weights. By construction, α^i is sparse since z^i is generated from the Beta process, which is reviewed in the next paragraph. This is different from the common Laplace prior [19], which leads to many small but often nonzero coefficients. Weight s^i and noise ϵ^i follow normal distributions parametrized by γ_s and γ_ϵ, respectively. Since there is full conjugacy in the hierarchical model, inference from given observations of x^i can be quickly computed by variational Bayesian [20] or Gibbs-sampling analysis, with analytic update equations.

The two-parameter beta process (BP) was introduced in [18]. Although it is allowed for $K \to \infty$, we assume a finite K for simplicity in this review. Let $[0, 1]$ be evenly divided into K bins. The kth bin is denoted by the interval $B_k = (\frac{k-1}{K}, \frac{k}{K}]$. For each k, we sample $\pi_k \sim \text{Beta}(\frac{a}{K}, \frac{b(K-1)}{K})$, where such a beta

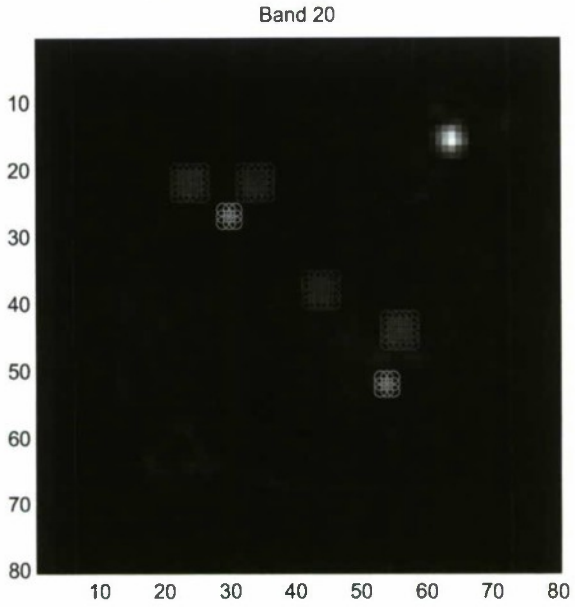
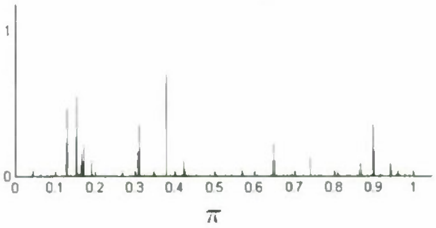
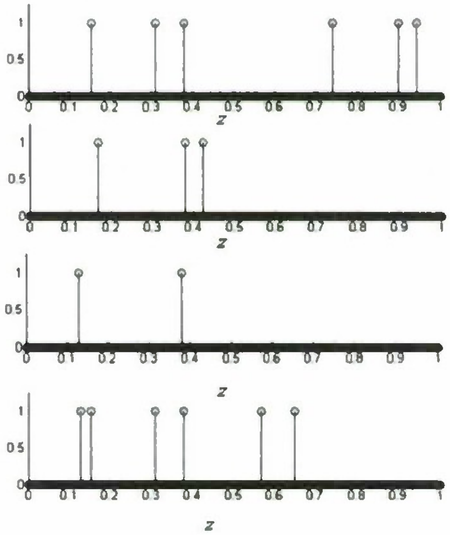


Figure 9: Ground truth locations of the 6 pieces of cloth.

distribution takes values over [0, 1] and has a U-shaped probability density function. Using π_k and B_k, we generate a new process $H(B) := \sum_{k=1}^{K} \pi_k \delta_{B_k}(B)$, where $\delta_{B_k}(B)$ equals one if $B = B_k$ and zero otherwise. In other words, $\pi_k = H(B_k)$, $k = 1, \ldots, K$, are a series of numbers between 0 and 1. The left plot below shows an example from [18] where $K = 1000$. Except for a few nonzeros, most locations of π are zero.





Then, π_k's are used as parameters to sample a series of 0/1 Bernoulli numbers $z_k \sim \text{Bernoulli}(\pi_k)$, i.e., $z_k = 1$ with probability π_k and $z_k = 0$ with probability $1 - \pi_k$. Four independent samples of z are shown in the right plot above, which were drawn with the same π shown in the left plot. Obviously, they are sparse, and there are repeats of 1's at the locations where π_k's are relatively large. By marginalizing π, it can be shown

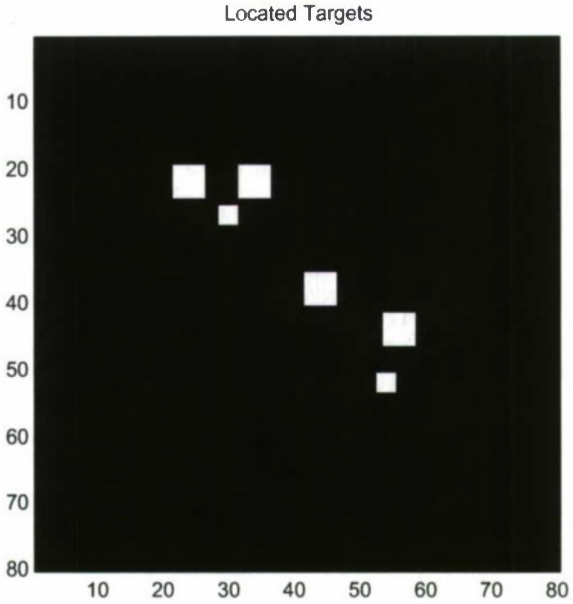


Figure 10: Found locations of the 6 pieces of cloth.

that the total number of nonzeros $\|z\|_0$ of z follows a Poisson distribution with parameter a/b. Furthermore, for any set of M vectors $\{z^1, \ldots, z^M\}$, the number of *unique* locations of 1's follow the Poisson distribution with the parameter $\sum_{i=1}^{N} \frac{a}{b+i-1}$. To have the desired jointly sparse α^i's, one shall adjust a and b.

4.4 Hyperspectral Image Reconstruction and Denoising

We apply the model $x^i = D\alpha^i + \epsilon^i$ to a hyperspectral image, where x^i's and the atoms in D are small 3D blocks (corresponding to patches for 2D images). Take a 150 × 150 × 210 hypercube for example, where 210 is the number of bands. A block, for instance, can be 3 × 3 × 210. The set of atoms contains multiple such blocks, each forming a vector d^m. The entire hyperspectral cube is decomposed to *overlapping* 3 × 3 × 210 blocks, each being a x^i. As such, all blocks of the hyperspectral image are modeled as linear combinations of the same set of atoms, and their combination coefficients are jointly sparse to a large extent. When there are missing voxels, the corresponding rows of $x^i = D\alpha^i + \epsilon$ are also missing. In case the observations are noisy, the hyper-prior γ_ϵ of ϵ is drawn from a non-informative gamma distribution.

To recover the hyperspectral image, Gibbs inference is applied to all blocks $x^i = D\alpha^i + \epsilon^i$ from given incomplete observations. Note that the atoms in D, all α^i, as well as noise ϵ^i are simultaneous inferred. The computation is relatively fast using the analytic inference update equations, which exist due to full conjugacy.

Numerical Results. We tested the above algorithm on a subset of the URBAN data. A reconstruction using 5% of the 150x150 spatial dimensions and 163 spectral bands was performed. Figures 14 and 15 show the results of the reconstruction. To check the quality of the reconstruction, we ran our target detection algorithm on the reconstructed hypercube. A comparison is shown in Figure 16.

4.5 Integrated Endmember Recovery and Hyperspectral Image Reconstruction

This is a project that we have partially done and will continue in the next phase of this contract. The above model $x^i = D\alpha^i + \epsilon^i$, which uses image patches as dictionary atoms, does not take advantage of the fact

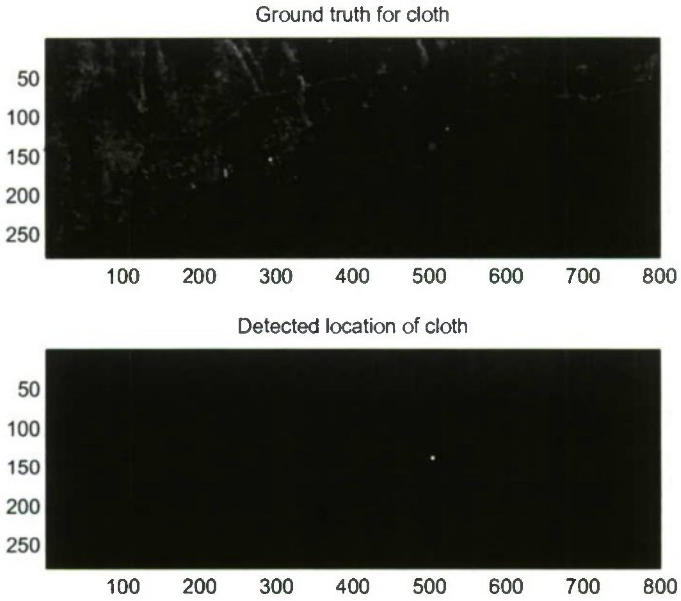


Figure 11: The top plot shows the whole data set and the ground truth location of the cloth(red), and the bottom shows the result of the detection of the cloth.

that all hyperspectral vectors are generated from a small number of endmembers. This is another nature of sparsity. We have been looking for ways to exploit this property for better performance.

The new model is $x^j = E\beta^j + \epsilon^j$ for each spatial pixel j and $\beta^i = \tilde{D}\tilde{\alpha}^i + \tilde{\epsilon}^i$ for each patch i of β. In the first set of equations, E is a dictionary of endmembers that form all the hyperspectral vectors. x^j is the hyperspectral vector at pixel j, which is assumed to be a linear combination of the endmembers, and ϵ^j is the error. In reality, the combination is nonlinear due to absorption, etc. This will be considered in ϵ^j as a part of the future work. The set of coefficients $\{\beta^j\}$ form another 3D cube, which has the same number of spatial pixels as the original hyperspectral cube but a much smaller number of layers (the 3rd dimension) equal to the number of endmembers. In other words, the first set of equations relate the larger 3D cube x to the smaller 3D cube β through endmembers. Then, in the second set of equations above, the dictionary model is applied to β. Each β^i is not a vector but a cube-like patch. It is assumed to be a sparse linear combination of a set of atoms given in the dictionary $\tilde{D}$. The dictionary $\tilde{D}$, coefficients $\tilde{\alpha}^i$, and errors $\tilde{\epsilon}^i$ are to be inferred from a given observation in x. In other words, what is described in the previous subsections is applied to β.

Preliminary Numerical Results. Eventually, we plan to learn the endmembers E from the data at the same time $\tilde{D}$, $\tilde{\alpha}$, and $\tilde{\epsilon}$ are learned. Modeling nonlinearity is also in our agenda. Our preliminary results are based on approximate endmembers, which are obtained using the algorithm VCA [21]. So at this moment, E is learned offline and not state-of-the-art.

In the test, the original cube is 150 × 150 × 163. VCA gave 6 endmembers, so β is 150 × 150 × 6. Two sets of images are shown in Figure 17 (a) and (b), corresponding to spectral bands 1 and 6, respectively. The first columns show the sample points on bands 1 and 6, which are smeared by noise (SNR 7.0685dB). The recovered cube has an SNR 14.8368dB. In other words, not only the cube is recovered from very few samples, but the SNRs are doubled. At this time, the work uses approximate endmembers. We plan to integrate endmember recovery and target detection seamlessly into this computation.

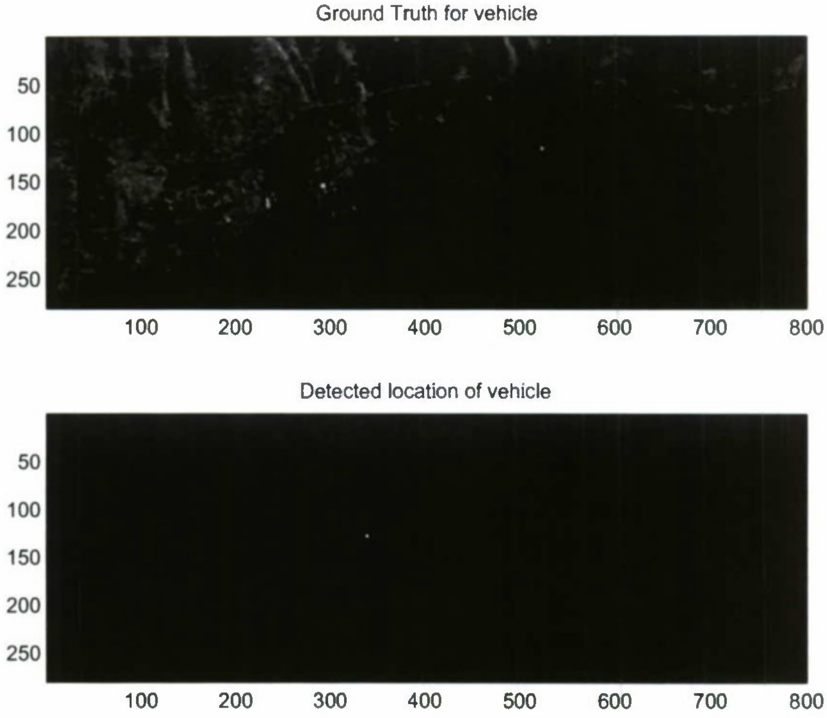


Figure 12: The top plot shows the whole data set and the ground truth location of the vehicle(red), and the bottom shows the result of the detection of the vehicle.

References

[1] Z. Guo, T. Wittman, and S. Osher, "L1 unmixing and its application to hyperspectral image enhancement," *UCLA CAM Report* **09-30**, 2009.

[2] T. Goldstein and S. Osher, "The split Bregman algorithm for L1 regularized problems," *UCLA CAM Report 08-29* , 2008.

[3] A. Szlam, "Private communication," 2009.

[4] NASA, "Aviris standard data product of moffett field, ca.," *Jet Propulsion Laboratory. [Online]* , p. ftp://137.78.180.72/pub/outgoing/stddata/f970620t01p02 r03.rfl.tar.gz, 2007.

[5] "http://www.agc.army.mil/hypercube/index.html,"

[6] S. Osher, M. Burger, D. Goldfarb, J. Xu, and W. Yin, "An iterative regularization method for total variation-based image restoration," *SIAM Journal on Multiscale Modeling and Simulation* **4**(2), pp. 460–489, 2005.

[7] Y. Yin, S. Osher, D. Goldfarb, and J. Darbon, "Bregman iterative algorithms for l1-minimization with applications to compressed sensing," *SIAM J. Imagins Sciences* **1(1)**, pp. 143–168, 2008.

[8] J. Cai, S. Osher, and Z. Shen, "Linearized bregman iterations for compressed sensing," *UCLA CAM Report* **08-06**, 2008.

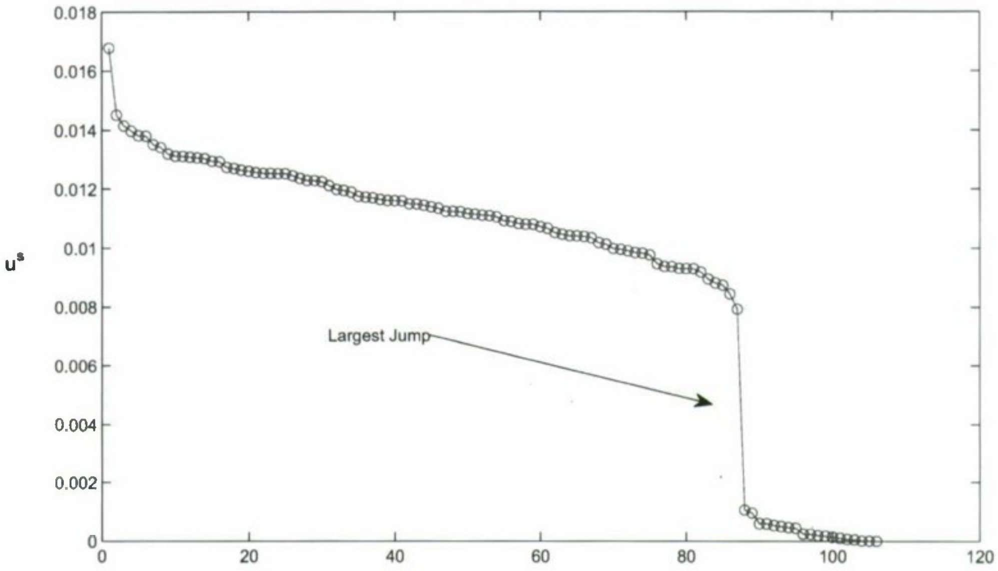


Figure 13: Sorted values of $u > 0$ for the target detection of gravel. It is clear where the largest jump is, and what the threshold should be.

[9] J. Cai, S. Osher, and Z. Shen, "Convergence of the linearized bregman iteration for l1-norm minimization," *UCLA CAM Report* **08-52**, 2008.

[10] S. Osher, Y. Mao, B. Dong, and W. Yin, "Fast linearized bregman iteration for compressive sensing and sparse denoising," *Communications in Mathematical Sciences* , 2008.

[11] W. Yin, "Analysis and generalizations of the linearized bregman method," *UCLA CAM Report* **09-42**, 2009.

[12] "http://dirsapps.cis.rit.edu/blindtest/,"

[13] M. Zhou, H.-G. Chen, J. Paisley, L. Ren, G. Sapiro, and L. Carin, "Non-parametric bayesian dictionary learning for sparse image representations," *NIPS* , 2009.

[14] B. A. Olshausen and D. J. Field, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," *Nature* **381**(6583), pp. 607–609, 1996.

[15] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Transactions on Signal Processing* **54**, 2006.

[16] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Transactions on Image Processing* **15**, 2006.

[17] J. Mairal, M. Elad, and G. Sapiro, "Sparse representation for color image restoration," *IEEE Transactions on Image Processing* **17**, 2008.

[18] J. Paisley and L. Carin, "Nonparametric factor analysis with beta process priors," *In Proc. of International Conference on Machine Learning* , 2009.

[19] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of Royal Statistical Society, Series B* **58**, 1994.

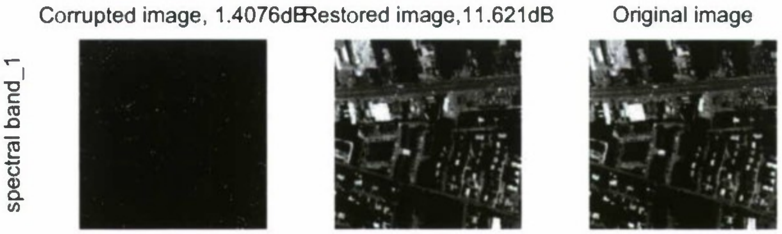


Figure 14: A plot of band 1 of the corrupted data, reconstructed data, and original data.

[20] M. J. Beal, "Variational algorithms for approximate bayesian inference," *PhD thesis, Gatsby Computational Neuroscience Unit, University College London* , 2003.

[21] J. M. P. Nascimento and J. M. B. Dias, "Vertex component analysis: a fast algorithm to unmix hyperspectral data," *IEEE Geoscience and Remote Sensing Letters* **43**(4), pp. 898–910, 2005.

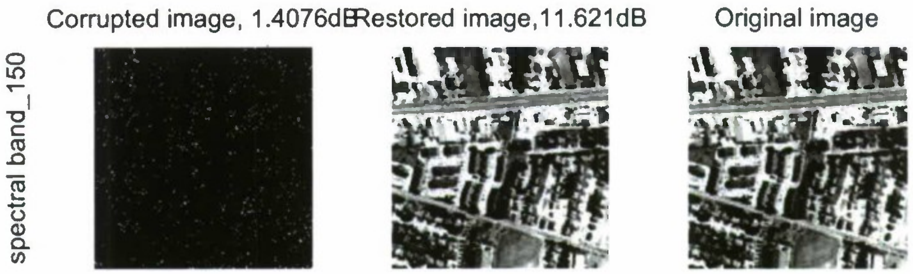


Figure 15: A plot of band 150 of the corrupted data, reconstructed data, and original data.

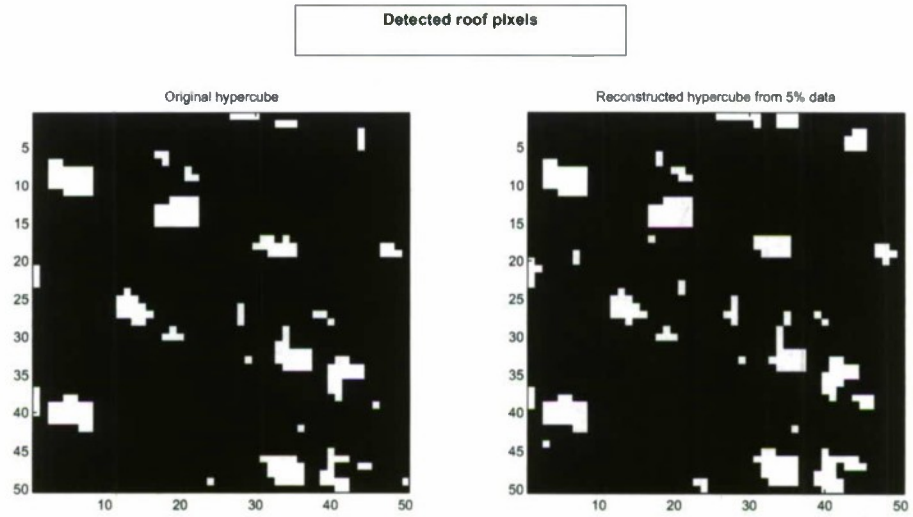
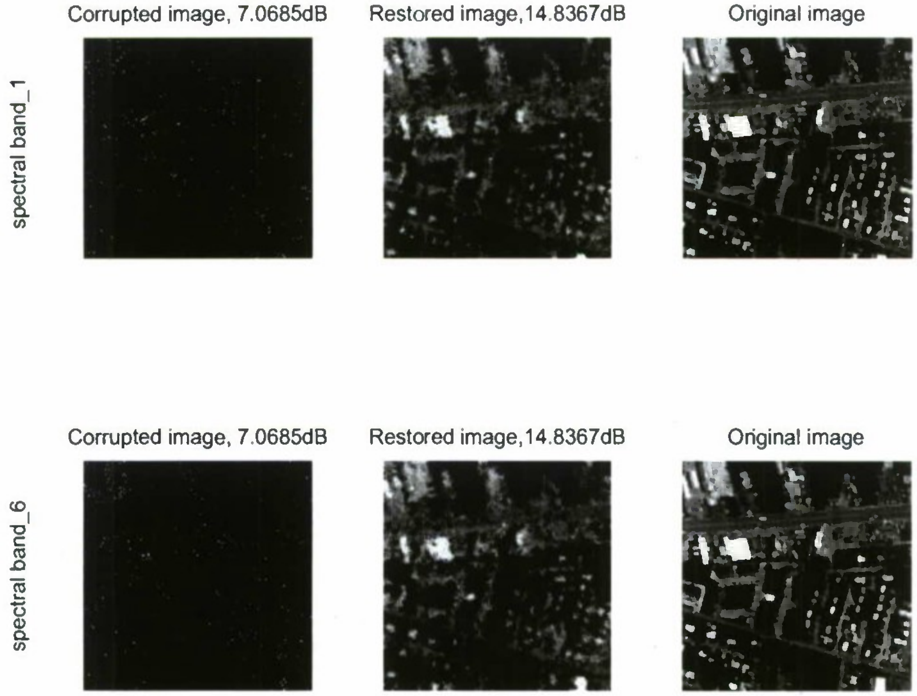


Figure 16: A comparison of targets detected using the original hypercube and the reconstructed hypercube.



(b) Band 6

Figure 17: Left to right: Very noise samples, recovered, and original. Note that the sample points (left column) themselves have a low SNR 7.0685dB. The recovered cube (middle column), including all voxels, has more than doubled SNRs.